

## **Instruction of RDA factory section data**

## Catalog

1. Instruction of RDA factory section data .....	3
1.1 Brief introduction of factory data .....	3
1.2 Structure of RDA factory data .....	3
1.3 Illustration of Phase Check.....	5

# 1. Instruction of RDA factory section data

## 1.1 Brief introduction of factory data

Factory data is mainly used in production line test to control the quality of products and technological process. Such as the serial number of phone and phase check information are always stored in factory section. The structure of factory section can be defined by customer, according to the requirements of their own product and production line. The factory section is usually located in the last section of flash, and the size is 8Kbytes.

## 1.2 Structure of RDA factory data

The structure of RDA 0xFAC00200 version factory data divide the 8Kbyte flash area into 8 equal units. The detailed structure of each unit is as follow:

```
#define TGT_FACTORY_BLOCK_SIZE 1024

#define FACTORY_VERSION                (0xFAC00200)
#define FACTORY_BLOCK_SIZE            (1024)
#define FACT_SIM_COUNT                (4)
#define FACT_IMEI_LEN                  (9)
#define FACT_SN_LEN                    (64)
#define FACT_MAC_LEN                   (6)
#define FACT_BT_MAC_FLAG               (0X1D0E)
#define FACT_STATION_COUNT             (15)
#define FACT_STATION_NAME_LEN          (12)
#define FACT_STATION_DESC_LEN          (32)
#define FACT_SUPPLEMENTARY_LEN         (544)
#define FACT_DEVICENAME_LEN            (32)
#define FACT_PINCODE_LEN               (16)

typedef UINT8 FACT_IMEI_T[FACT_IMEI_LEN];

typedef UINT8 FACT_SN_T[FACT_SN_LEN];

typedef struct {
    UINT16                activated;        //0x00000000
    UINT8                 mac[FACT_MAC_LEN]; //0x00000002
} FACT_MAC_T; //Size : 0x8

typedef UINT8 FACT_STATION_NAME_T[FACT_STATION_NAME_LEN];
```

```

typedef struct {
    UINT16 activated;
    UINT8 dev_addr[FACT_MAC_LEN];
    UINT8 dev_name[FACT_DEVICENAME_LEN];
    UINT8 pin_code[FACT_PINCODE_LEN];
} TGT_BT_INFO_T;

typedef struct {
    UINT16 activated;
    UINT8 mac_addr[FACT_MAC_LEN];
    UINT8 mac_ap1[FACT_MAC_LEN];
    UINT8 mac_ap2[FACT_MAC_LEN];
    UINT8 mac_ap3[FACT_MAC_LEN];
    UINT8 reserved[2];
} TGT_WIFI_INFO_T;

typedef struct {
    UINT32 version; //0x00000000
    FACT_IMEI_T imeiSv[FACT_SIM_COUNT]; //0x00000004
    UINT8 mbSn[FACT_SN_LEN]; //0x00000028
    UINT8 mpSn[FACT_SN_LEN]; //0x00000068
    TGT_BT_INFO_T btInfo; //0x000000A8
    TGT_WIFI_INFO_T wifiInfo; //0x000000E0
    FACT_STATION_NAME_T stationNames[FACT_STATION_COUNT];
//0x000000FC
    UINT16 stationPerformed; //0x000001B0
    UINT16 stationPassed; //0x000001B2
    UINT8 stationLastDesc[FACT_STATION_DESC_LEN]; //0x000001B4
    UINT8 supplementary[FACT_SUPPLEMENTARY_LEN]; //0x000001D4
    UINT32 sequence; //0x000003F4
    UINT32 crc; //0x000003F8
    UINT32 crcInverted; //0x000003FC
} FACT_FACTORY_BLOCK_T; //Size : 0x400

```

Description of data structure:

**Version:** Magic number information. This number should be updated when the data structure changed.

**imeiSv:** IMEI number of the phone.

**mbSn:** bsn of the phone.

**mpSn:** psn of the phone.

**btInfo:** Information of Bluetooth device.

**wifiInfo:** Information of wifi device.

**stationNames:** Name of stations. Most support 15 stations, and this station is one to one correspondence with Bit0-Bit14 of stationPerformed and stationPassed. As follow:

stationNames[0]  $\longleftrightarrow$  stationPerformed Bit0, stationPassed Bit0

...

stationNames[14]  $\longleftrightarrow$  stationPerformed Bit14, stationPassed Bit14

**stationPerformed:** This station is checked or not, and “1” represent has been checked.

**stationPassed:** This station is checked pass or fail. “1” represent passed, and “0” represent failed.

This flag is used only when the stationPerformed is “1”.

**Supplementary:** Customization information.

**Sequence:** Block management serial number. Set bit[0] as “0” and other value will be considered invalid. The initial sequence is “0”, and the step is “2”.

**Crc:** The CRC of the whole module, and itself is not contained. The CRC is calculated begin the start of the module, and the length is BLOCK\_SIZE-8.

**crcInverted:** Bit reversal CRC, this can increase the robust performance of CRC.

## 1.3 Illustration of Phase Check

[Assume]: There are four stations to be checked, the test sequence as: DOWNLOAD, WRITESN, CFT, IMEI.

stationPerformed=0xFFF7, stationPassed=0xFFF3

[Analysis]: The corresponding relation of stationPerformed, stationPassed and check station is list as follow:

BIT0: DOWNLOAD

BIT1: WRITESN

BIT2: CFT

BIT3: IMEI

stationPerformed = 0xFFF7, represent the DOWNLOAD, SN, CFT test station are passed, but the IMEI test is failed.

stationPassed=0xFFF3, represent the DOWNLOAD, SN test station are passed, but the CFT test is failed.